Self-Planning Code Generation with Large Language Models

Year: 2023 **|** *Citations: 174* **|** *Authors: Xue Jiang, Yihong Dong, Lecheng Wang*

Abstract

Although large language models (LLMs) have demonstrated impressive ability in code generation, they are still struggling to address the complicated intent provided by humans. It is widely acknowledged that humans typically employ planning to decompose complex problems and schedule solution steps prior to implementation. To this end, we introduce planning into code generation to help the model understand complex intent and reduce the difficulty of problem-solving. This paper proposes a self-planning code generation approach with large language models, which consists of two phases, namely planning phase and implementation phase. Specifically, in the planning phase, LLM plans out concise solution steps from the intent combined with few-shot prompting. Subsequently, in the implementation phase, the model generates code step by step, guided by the preceding solution steps. We conduct extensive experiments on various code-generation benchmarks across multiple programming languages. Experimental results show that self-planning code generation achieves a relative improvement of up to 25.4% in Pass@1 compared to direct code generation, and up to 11.9% compared to Chain-of-Thought of code generation. Moreover, our self-planning approach also enhances the quality of the generated code with respect to correctness, readability, and robustness, as assessed by humans.